Setting up Firebase Core & Realtime Database

Resources:

<https://github.com/myflashlab/Firebase-ANE/wiki>

<https://github.com/myflashlab/Firebase-ANE/wiki/A.-Get-Started>

<https://github.com/myflashlab/Firebase-ANE/wiki/B.-Realtime-Database>

First set up a new firebase project. By setting up the Firebase project you are setting up the database that you will be able to call from and send to… to set up the project goto

<https://console.firebase.google.com/?pli=1>

aroutine-644fc

Once you click on “create” you will enter the Welcome to Firebase page.

At the top of the page in the blue section are 3 circles. The middle circle is used to

“Add Firebase to your Android app”… click on this choice.

To Register app follow this naming convention

Android package name: air.com.site.”YourAppName”

App nickname: ~~~Optional~~~

Download google-services.json

Then add the json to the root directory of your app (where the .fla is)

After completing this it is time to set up your Firebase ANE’s. If you haven’t yet done so, create an ANE folder in you app to store all the main and dependent ANE files for your project. Once you have that you will need to put the firebaseCore.ane file within this folder….

… But … you will notice that you don’t have and can’t find a copy of the firebaseCore.ane file. You will need to generate the ANE file with the use of the AneMaker-V5.1.1.air Installer Package that was included in the myFlashLabs / ANEs / firebase / core folder that Ellertson provided. To use the AneMaker you need to have already set up, downloaded and placed your google-services.json because you will need to add your Android package name and where the .json is located. Then just click the “Generate ANE” button and it will create the firebaseCore.ane file you need. (Note… don’t worry about it asking for the GoogleService\_Info.plist file, this is only for iOS apps).

Place the firebaseCore.ane file into the ANE folder of you app

To run Firebase Core you need the following dependencies….

| **On the Android side** | **On the iOS side** |
| --- | --- |
| androidSupport.ane V26.0.2 | overrideAir.ane V5.1.0 |
| overrideAir.ane V5.1.0 | FirebaseAnalytics.framework |
| firebase\_common.ane V11.6.0 | FirebaseCore.framework |
| firebase\_iid.ane V11.6.0 | FirebaseCoreDiagnostics.framework |
| googlePlayServices\_base.ane V11.6.0 | FirebaseNanoPB.framework |
| googlePlayServices\_basement.ane V11.6.0 | FirebaseInstanceID.framework |
| googlePlayServices\_tasks.ane V11.6.0 | GoogleToolboxForMac.framework |
| googlePlayServices\_appinvite.ane V11.6.0 | nanopb.framework |

These dependency ane’s are also not included in your myFlashLab’s folder, so you will need to download them from <https://github.com/myflashlab/common-dependencies-ANE/tree/f43d0e760a79efcf36ed6846df50bce93dfee0b4>

I recommend just downloading them all, so that if you need a dependency for another ANE you will hopefully now have it.

Place the dependency ANE’s into your ANE folder and then it is time to load the ANE’s into your flash project.

In Flash, click off the stage so that you bring up the Properties menu. Under the Publish tab, click on the wretch icon next to the Script heading. This will pull up the Advanced ActionScript 3.0 Settings window. Click on the Library Path tab which is where you will link your ANE files. Click on the Browse to ANE file button and then make your way to your ANE folder and one at a time add them to your .fla file.

Once this is all done you need to go into your Publish Setting and then your AIR for Android Settings. Then click on the Permissions tab and select the following Permissions:

INTERNET

WAKE\_LOCK

ACCESS\_NETWORK\_STATE

Make sure that you are testing your app on your device.

Then run your .fla to set up your .xml manifest with all the current information….

Open you manifest .xml file and add / verify the following code…. Make sure to pay attention to the areas that say to change the first part of some of the code to your Android Package Name that you created during the Google Firebase setup (above)

<!--

FOR ANDROID:

-->

<manifest android:installLocation="auto">

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"/>

<uses-permission android:name="android.permission.WAKE\_LOCK"/>

<!--

Required by firebase\_iid.ane

Change "air.com.doitflash.firebaseCore" to your own app package name

-->

<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE" />

<permission android:name="air.com.doitflash.firebaseCore.permission.C2D\_MESSAGE" android:protectionLevel="signature" />

<uses-permission android:name="air.com.doitflash.firebaseCore.permission.C2D\_MESSAGE" />

<application>

<activity>

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

<intent-filter>

<action android:name="android.intent.action.VIEW" />

<category android:name="android.intent.category.BROWSABLE" />

<category android:name="android.intent.category.DEFAULT" />

<!-- Change "air.com.doitflash.firebaseCore" to your own app package name -->

<data android:scheme="air.com.doitflash.firebaseCore" />

</intent-filter>

</activity>

<!--

Required by the firebase\_common.ane

Change "air.com.doitflash.firebaseCore" to your own app package name

-->

<provider

android:name="com.google.firebase.provider.FirebaseInitProvider"

android:authorities="air.com.doitflash.firebaseCore.firebaseinitprovider"

android:exported="false"

android:initOrder="100" />

<service

android:name="com.myflashlab.firebase.core.MyFirebaseInstanceIdService"

android:exported="true">

<intent-filter>

<action android:name="com.google.firebase.INSTANCE\_ID\_EVENT"/>

</intent-filter>

</service>

<!-- Required by the googlePlayServices\_basement.ane -->

<meta-data

android:name="com.google.android.gms.version"

android:value="@integer/google\_play\_services\_version" />

<!--

Required by firebase\_iid.ane

Change "air.com.doitflash.firebaseCore" to your own app package name

-->

<receiver

android:name="com.google.firebase.iid.FirebaseInstanceIdReceiver"

android:exported="true"

android:permission="com.google.android.c2dm.permission.SEND" >

<intent-filter>

<action android:name="com.google.android.c2dm.intent.RECEIVE" />

<category android:name="air.com.doitflash.firebaseCore" />

</intent-filter>

</receiver>

<receiver android:name="com.google.firebase.iid.FirebaseInstanceIdInternalReceiver" android:exported="false" />

<service android:name="com.google.firebase.iid.FirebaseInstanceIdService" android:exported="true">

<intent-filter android:priority="-500">

<action android:name="com.google.firebase.INSTANCE\_ID\_EVENT" />

</intent-filter>

</service>

<!-- Required by googlePlayServices\_base.ane -->

<activity android:name="com.google.android.gms.common.api.GoogleApiActivity"

android:theme="@android:style/Theme.Translucent.NoTitleBar"

android:exported="false"/>

</application>

</manifest>

<!--

FOR iOS:

-->

<InfoAdditions>

<!--iOS 8.0 or higher can support this ANE-->

<key>MinimumOSVersion</key>

<string>8.0</string>

</InfoAdditions>

<!--

Embedding the ANE:

-->

<extensions>

<!--

Download the dependency ANEs from https://github.com/myflashlab/common-dependencies-ANE

-->

<extensionID>com.myflashlab.air.extensions.dependency.firebase.common</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.firebase.iid</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.googlePlayServices.appinvite</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.googlePlayServices.base</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.googlePlayServices.basement</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.googlePlayServices.tasks</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.androidSupport</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.overrideAir</extensionID> <!-- Required for iOS and Android -->

<!-- And finally embed the Firebase core ANE -->

<extensionID>com.myflashlab.air.extensions.firebase.core</extensionID>

</extensions>

-->

Once this is complete it is time to set up Firebase Core in your ActionScript code. To do this you only need to do two things…

Import to AS3 code with …

import com.myflashlab.air.extensions.firebase.core.Firebase;

then just call it… in your constructor using….

Firebase.init();

\*\*\*Note….  If you are trying to enable DynamicLinks in your app, you need to pass true to the Firebase init method. Firebase.init(true);

Once this is done… then you need to install Firebase Realtime Database

Add the firebase realtime database ANE’s to your .fla

You need

Firebase\_database.ane

Firebase\_databaseConnection.ane

firebaseDatabase.ane

Add these ANE’s to your .fla just like the previous ANE’s. Also, verify that you permissions now include:

c2dm.permission.RECEIVE

ne.permission.C2D\_MESSAGE

In addition your extensions list should now look like….

<extensions>

<extensionID>com.myflashlab.air.extensions.firebase.core</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.androidSupport</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.firebase.common</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.firebase.iid</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.overrideAir</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.googlePlayServices.base</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.googlePlayServices.basement</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.googlePlayServices.tasks</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.googlePlayServices.appinvite</extensionID>

<extensionID>com.myflashlab.air.extensions.firebase.realtimeDB</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.firebase.database.connection</extensionID>

<extensionID>com.myflashlab.air.extensions.dependency.firebase.database</extensionID>

</extensions>

In your Actionscript file you need to add:

DB.init();

Add this after where you put the Firebase.init();

Now you wont be able to post anything to your database until you once again go back to that database and in the Database Rules make the following changes

{

"rules": {

".read": true,

".write": true

}

}

And while this is now not secure… this is the best we can do until Firebase Authentication is set up.

Now in your code you can push data to the database….

var myRef:DBReference = DB.getReference("message");

myRef.setValue("Your\_Message\_Goes\_Here");

myRef.addEventListener(DBEvents.VALUE\_CHANGED, onDataChange);

function onDataChange(e:DBEvents):void

{

// This method is called once with the initial value and again

// whenever data at this location is updated.

if (e.dataSnapshot.exists)

{

if (e.dataSnapshot.value is String) trace("onValueChanged String value = " + e.dataSnapshot.value);

else if (e.dataSnapshot.value is Number) trace("onValueChanged Number value = " + e.dataSnapshot.value);

else if (e.dataSnapshot.value is Boolean) trace("onValueChanged Boolean value = " + e.dataSnapshot.value);

else if (e.dataSnapshot.value is Array) trace("onValueChanged Array value = " + JSON.stringify(e.dataSnapshot.value));

else trace("onValueChanged Object value = " + JSON.stringify(e.dataSnapshot.value));

}

}

Now this code will just send the message to the database and if you run this code again with a different message it just overwrites the previously sent data.

Still to figure out….

Send data without overwriting

Get data from the database